

U. S. PATENT APPLICATION

of

DENIS BEAULIEU

for

METHOD AND APPARATUS FOR LOOP DETECTION AND
DISSOLUTION IN A COMMUNICATION NETWORK

METHOD AND APPARATUS FOR LOOP DETECTION AND DISSOLUTION IN A COMMUNICATION NETWORK

BACKGROUND OF THE INVENTION

The present invention relates to a method of loop detection and dissolution in a communication network including a plurality of nodes or network elements. More specifically, the present invention relates to an arboreal routing scheme for delivering communications to all nodes in the system without creating a loop.

Certain communication systems include a voice party line on every network element or node located within a plural node communication system. The voice of each participant in the party line is sampled and put in packets. These voice packets are distributed from the originating nodes to the nodes that are directly or indirectly connected to the originating node. At each node, the voice packet received from the each of the directly connected nodes is summed with the voice packets from all other nodes. This process is sometimes referred to as processing the "digital bridge".

Voice packets are not sent back on the same port on which the voice packets have been received to avoid an echo effect. However, under certain conditions a loop forms within the network topology. When there is a loop within the network, a voice sample will be summed with itself repeatedly. The present

invention provides a method of detecting and eliminating a loop which has formed within the network.

In addition, the present invention provides an arboreal routing protocol which detects the presence of a loop in the communication network and eliminates the detected loop without damaging the integrity of the communication of the nodes within in the network. Moreover, the arboreal routing protocol of the present invention requires little computing power from the network elements and converges very rapidly to a solution.

BRIEF DESCRIPTION OF THE DRAWINGS

The exemplary embodiments of the present invention will be described with reference to the accompanying drawings.

Fig. 1 is a schematic drawing of a plural node communications network showing the arboreal routing for a message originating at NE1;

Figs. 2 is a schematic drawing of a communications network illustrating an exemplary arboreal routing during the voting state;

Figs. 3 and 4 are schematic drawings of a communications network illustrating an exemplary arboreal routing during the elected loop-checking state; and

Figs. 5a and 5b are schematic drawings of the network performing a topological transformation on a node having more than three links.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

Fig. 1 illustrates an exemplary embodiment of an arboreal routing for a broadcast frame originating at node NE1. Each of the plurality of nodes NE1-NE12 are linked through ports to at least one neighbor node and indirectly linked through a neighbor node to at least one remote node.

A neighbor node which is directly connected and in communication with another node is referred to as a "Fully Qualified Neighbor" (FQN). However, if two nodes are connected to one another, but the communication is unidirectional between the two nodes, then these nodes would not be considered FQN. Different nodes within the system may each have a different number of FQNs.

Fig. 2 schematically illustrates an exemplary embodiment wherein a plurality of nodes A1 through A9 are connected to one another through various ports 1-10. For example, in Fig. 2a, node A1 has two FQNs, i.e. nodes A5 and A2, and node A2 has three FQNs, i.e. nodes A1, A3 and A6.

When a change in topology of the network is detected or an external event occurs within the network, then at least one of the nodes in the network sends a vote token to each of its FQNs. Assuming a node detects a change in the topology of the network, then that node will send a vote token to its FQNs. The vote token contains the address of the node from which the token originated and the number of FQNs of the originating node.

If the node that sent the vote token does not receive it back within a predetermined time, preferably one second, it will un-mute all of its links and send a new set of vote tokens. This procedure is intended to avoid the isolation of any node.

Each node receiving the vote token from the originating node will compare the number of its FQNs with the number of FQNs of the originating node recorded on the received vote token. If the receiving node has the same or less FQNs than the originating node, the receiving node forwards the received vote token to all of the receiving node's FQNs, with the exception of the node from which the token was received.

If the receiving node has more FQNs than the originating node, then the receiving node "swallows" the token from the originating node and sends a new vote token to all of its FQNs with the exception of the node from which the swallowed token was received. The new vote token includes the identification and number of FQNs of the node by which originated.

This procedure is repeated at each node in the communication system until at least one of the nodes receives a vote token which that node had originated. Therefore, more than one node may be elected as the loop checker in a network.

The return of a vote token to its originator indicates that there is a loop within the network. Furthermore, the network determines which node within the loop as the highest number of FQNs, and may therefore have one of its ports "muted" without isolating that node from the other nodes in the network.

The originating node which receives the returned vote token becomes the elected-loop checker. It should be noted more than one node may become an elected-loop checker. In the exemplary embodiment, it is preferred that only one node at a time be performing the loop check. There are two mechanism which that insure that only one elected loop checker checks for loops at the same time. One mechanism being that each elected-loop checker performs a random delay so that the loop check tokens are sent at different times. Second, if a loop-checker node receives a loop-check token from another loop-checker node, it does not retransmit the token and the loop-checker node stops checking for loops. After a random delay, the loop-checker node transmits new loop-check tokens.

In the exemplary embodiment, the elected-loop checker randomly delays sending out the loop-check token. The random delay can vary in time, but in a preferred embodiment the random delay is anywhere from 1 second to 20 seconds. During the random delay, if the elected-loop check node receives a loop-check token from of another node, the elected loop check node will forward up to two loop-check tokens. If more than two loop-check tokens are received from the same node, the node will swallow these loop-check tokens. The reason for not allowing all loop-check tokens of other nodes to be forwarded during the random delay is the probability that these loop-check token(s) originated from another closed loop within the network.

Once the random delay is finished, the elected-loop check node sends a loop-check token to each of its non-muted links which are linked to a FQN. The

loop-check token identifies the elected loop-check node and the port through which it was sent. Each of the FQNs that receive the loop-check token forward the loop-check token to their respective FQNs, with the exception of the node from which the loop-check token was received. This process continues until the loop-check token returns back to the elected loop-checker node through a port other than the port through which it was sent.

The elected loop-check node will not retransmit vote tokens or loop check tokens during non-random delay periods to insure that only one node in each loop should be in the loop-checking state. If more than one node in each loop is in a loop-checking state then there is a chance that two ports are being muted simultaneously or nearly simultaneously, therefore isolating certain nodes or network elements from the other members within the network.

Once a loop-check token has returned to the elected loop-checker node from which the token originated, the port which delivered the loop-check token back to the elected loop-checker node is then "muted", opening the loop, and the other ports of the elected loop-checker node are "un-muted".

By way of example and referring to Fig. 2A, it is assumed that node A1 detects a change in topology of its neighbors. Upon detecting a change in the topology, node A1 sends a vote token containing its node name and the number of FQNs, i.e. 2. In this example, the name of the token will be V.A1.2. The V.A1.2 token is sent by link 1 and 2. V.A1.2 token sent through link 2 reaches node A2

which has three FQNs. Because node A2 has more FQNs than node A1, node A2 swallows V.A1.2 token.

A2 then sends a new vote token V.A2.3 through links 4 and 6. It does not send the new vote token through link 2 because link sent the swallowed vote token. V.A2.3 token will contain the node name and number of FQNs for node A2. It should be noted, that in the exemplary embodiment, the vote tokens are allowed to pass through muted links. V.A2.3 token sent through link 6 will travel the following path: A6, A7, and A8. Node A9 will receive a copy of the vote token, V.A2.3, which node A9 will swallow due to fact that there are no other neighbors for node A9 to forward the token. Node A3 will duplicate the V.A2.3, and send one of the copies back to node A2.

Upon receiving its won vote token, A2 is elected as a loop checker.

Vote token V.A1.2 that is sent through link 1 travels the following path: node A5, A4 and A3, respectively. Because node A3 has 3 FQNs and node A1 has 2, node A3 swallows the copy of the V.A1.2 token and sends a new vote token, V.A3.3 through links 4 and 8. Eventually, up to 4 copies of V.A3.3 token return to node A3 electing it as a loop checker.

At the end of the voting phase we have A2 and A3 elected as the loop checkers.

After a random delay, the elected loop checker A2 sends two loop check tokens which contain the node name and the address of the port which they were sent. In an exemplary embodiment, the random delay is about 1 to 20 seconds. In

this example, we will name the loop check token sent by node A2 through port 2 as C.A2.2 token, and the loop check token sent by node A2 through port 4 as C.A2.4. It should be noted that in this exemplary embodiment, unlike the vote tokens, loop check tokens cannot pass through ports which are muted, e.g. link 6. Otherwise, the loop check tokens will travel through in the network.

As shown in Fig. 3, assuming that C.A2.2 returns back to node A2 by port 4 before the other token arrives to A2, then node A2 will unmute all its links and then mute link 4. Once this process is finished A2 is no longer a loop checker and it will pass the C.A2.4 token to its new unmuted links. Because link 4 is now muted, C.A2.4 will not pass through link 4 and will be swallowed the second time it reaches either the A2 node or the A3 node.

A3 is then the only remaining loop checker node. After a random delay, node A3 sends two loop check tokens that contain the node name and the address of the port by which the token was sent. In this example, the loop check token sent by port 5 is C.A3.5 and the loop check token sent by port 8 is C.A3.8. As shown in Fig. 4, if C.A3.5 returns to node A3 through link 8, then node A3 will un-mute all its links and then mute link 8. If C.A3.8 token returns to node A3 through link 5, then node A3 will un-mute all its links and then mute link 5. Once this is done, node A3 is no longer a loop checker and the network topology does not contain a loop.

In another exemplary embodiment as shown in Figs. 5a and 5b, some of the nodes may have more than three links. It is preferred for the algorithm to function

properly that each of the nodes have at the most three links. When a node in the network has more than three links, the present invention performs a topological transformation. For example, Fig. 5a illustrates that node A has four links. In an exemplary embodiment, the present invention performs a topological transformation to make the node A which has four links appear as a two nodes have three links each as shown in Fig. 5b.

It may thus be seen from the foregoing that the above apparatus and method provide for an arboreal routing protocol that uses the most efficient route to send a broadcast frame. Because there are no loops, each frame is received only once by every node within the network.

Although the present invention and its advantages have been described in detail, it should be understood that various changes, substitutions and alterations can be made herein departing from the spirit and scope of the invention as defined by the appended claims.